AMENDMENTS TO THE DRAWINGS

The attached sheet of drawings includes changes to Figure 2. This sheet replaces the original sheet including Figure 2. In Figure 2, packet flow numbers labeled "PKT FLOW 1," "PKT FLOW 2," and "PKT FLOW S," have been shown in each of the instruction streams and as being employed by each of the cache miss predictors.

Attachment: Replacement Sheet

REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-20 are pending in the application. The Examiner additionally stated that claims 1-20 are rejected. By this amendment, claims 1, 3-13, 15, and 18-19 have been amended. Hence, claims 1-20 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Specification

The Examiner objected to the specification, noting that the title of the invention is not descriptive, and requiring a new title that is clearly indicative of the invention to which the claims are directed. Applicant has amended the title with a new title that clearly indicates the invention to which the claims are directed. Accordingly, it is requested that the objection to the specification be withdrawn.

In the Drawings

The Examiner objected to the drawings under 37 CFR 1.83(a), noting that the drawings must show every feature of the invention specified in the claims. In particular, the Examiner stated that the specifics of claims 16, 17, and 20 must be shown or the feature(s) cancelled. Applicant has amended the drawings in Fig. 2 to show packet flow numbers in each of the instruction streams as being provided to each of the predictors. This feature is recited in the specification page 15, lines 7-12. Accordingly, it is requested that the objection to the drawings be withdrawn.

In the Claims

Claim Objections

The Examiner objected to claim 6 because "steams" should be replaced with -- streams -- and objected to claim 11 because "stream" should be replaced with -- streams --. By this Amendment, these corrections are made, therefore, it is requested that the objections to claims 6 and 11 be withdrawn.

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-14 and 18-19 under 35 U.S.C. §102(b) as being anticipated by Yoaz et al., "Speculative Techniques for Improving Load Related Scheduling," May 1999 (hereinafter, "Yoaz"). In addition, the Examiner cited Parady, U.S. Patent 5,933,627 (as applied in the previous Office Action and hereinafter referred to as "Parady") as extrinsic evidence for showing that it is common to have separate hardware streams for each thread. Applicant respectfully traverses the Examiner's rejections.

Prior to providing a claim-by-claim analysis, a brief summary of the teachings of Yoaz and Parady are provided below, vis-à-vis the invention disclosed by Applicant in the instant application. This information is provided to aid the Examiner during reconsideration of the claims.

Yoaz discloses three techniques to address *instruction scheduler* limitations in an out-of-order engine, where the *instruction scheduler* is responsible for *dispatching instructions to execution units* based upon dependencies, latencies, and resource availability. The problem that is noted by Yoaz, and which motivates his *instruction scheduler* techniques, is that dynamic latencies of load instructions are unknown, so *scheduling* dependent instructions is based on either load-use delay or pessimistic delay. (cf. Abstract) Yoaz expands his teachings in the are of hit/miss predictions by noting that the new concept presented is to predict which loads will miss the cache, thus *delaying the dependent instructions* until the needed data is fetched. Yoaz opines that this increases performance directly by saving a few clocks through *the scheduling of* load-dependent instructions *to execute* at the exact time the data is retrieved. Yoaz's technique involves a hardware approach that is based on *a per-lad binary prediction* of a hit or miss in the cache. (cf. page 44, lines 8-27) The author specifically notes that once all instruction dependencies have been resolved, the *scheduler's* remaining task is to *dispatch instructions* in a way that maximizes the utilization of available resources, while minimizing instruction latencies. Further, Yoaz suggests dynamically predicting whether a specific load with hit or miss the cache, *to facilitate the scheduling operation.* (cf. page 46, lines 17-31) Yoaz

moreover intimates that multi-threading may benefit from hit/miss prediction, and that the prediction may be used to govern a thread switch if a load is predicted to miss the L2 cache. (cf. page 47, lines 25-29)

Parady discloses a method and apparatus for switching between threads of a program *in response to a long-latency event.* In one embodiment, the long-latency events are load or store operations which trigger a thread switch *if there is a miss* in the level 2 cache. In addition to providing separate groups of registers for multiple threads, a group of program address registers pointing to different threads are provided. A switching mechanism switches between the program address registers *in response to* the long-latency events. (cf. Abstract) Parady also defines the process whereby a multithreading processor interleaves threads in such a manner as described above (i.e., in response to a long-latency event) as "coarse-grain multithreading," (cf. col. 2, lines 8-10) and furthermore teaches the concept of "a switching mechanism [that] switches between the program address registers *in response to* the long-latency events. Parady discloses the switching mechanism as "[t]hread switching logic 112 provided to give a hardware thread switching capability. The indication that a thread switch is required is provided on a line 114 providing an *L2-miss indication from cache control/system interface 22.*" He further teaches that "*[u]pon such an indicatio*n, a switch to the next thread will be performed." (cf. col. 3, lines 57-62) Furthermore, in a background discussion of his invention, Parady cites an IBM article that distinguishes between processors to which his invention is directed (i.e., coarse-grain multithreaded processors) and fine-grain multithreaded processors, that is, those processors which interleave threads on a cycle-by-cycle basis. (cf. col. 2, lines 6-10) Parady's invention and disclosure is directed towards problems associated with coarse-grain processors: those processor that switch threads *in response to* long latency events.

In contrast to the teachings of Yoaz and Parady, Applicant's invention is directed towards a processor having multiple hardware streams supporting multiple data threads, and a data cache. In this processor, Applicant discloses a system for *fetching* instructions from up to P of the multiple hardware streams to a pipeline, where P is less than the number of multiple hardware streams. The system has a fetch stage, multiple hit/miss predictors,

and a fetch algorithm. The fetch stage is configured to *simultaneously fetch every cycle*, the instructions from up to the P of the multiple hardware streams. The multiple hit/miss predictors are each associated with a corresponding one of the multiple hardware streams, and are each configured to forecast, at the fetch stage, whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache. The fetch algorithm is coupled to the multiple hit/miss predictors. The fetch algorithm selects, on a cycle-by-cycle basis, the P of the multiple hardware streams from which to fetch the instructions.

Yoaz's technique involving hit/miss prediction is clearly directed towards *scheduling instructions for execution* which *have already been fetched into the pipeline.* Yoaz fails even to note the problem that is addressed by Applicant in the instant application.

By Parady's own admission, his invention is directed towards coarse-grain multithreading applications. In these applications and corresponding multithreaded processors, thread switches occur *in response to long-latency events.* Such a configuration is evidenced by signal line 114 in Parady that provides an L2-miss indication from cache control/system interface 22. By stating a specific multithreading application (i.e., coarse-grain multithreading) to which his invention is directed, in contrast to the multithreading application to which Applicant's invention is directed (i.e., fine-grain multithreading, that switches threads on a cycle-to-cycle basis), Parady thus teaches away from the present invention.

It therefore does not follow that Yoaz anticipates a system for *fetching* instructions into a multi-threaded processor pipeline, for his article is directed towards *execution scheduler* limitations. Furthermore, to suggest that the teachings of Parady and Yoaz can be combined in a manner relative to Applicant's invention is out of place because Yoaz teaches how to schedule instructions for execut ion and Parady's invention is directed towards *responding to the occurrence of long-latency events.*

In view of the above summarizations, a claim-by-claim analysis will now be presented.

Amended claim 1 is provided below for ease of reference.

1. In a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from up to P of the multiple hardware streams to a pipeline, where P is less than the number of multiple hardware streams, the system comprising:

a fetch stage, for simultaneously fetching every cycle, the instructions from up to the P of the multiple hardware streams;

multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast, at the fetch stage, whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache; and

a fetch algorithm, coupled to said multiple hit/miss predictors, configured to select, on a cycle-by-cycle basis, the P of the multiple hardware streams from which to fetch the instructions.

Claim 1 recites, in combination, within a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from up to P of the multiple hardware streams to a pipeline. The system has a fetch stage, for simultaneously fetching every cycle, the instructions from up to the P of the multiple hardware streams. The system also has multiple hit/miss predictors that are each associated with a corresponding one of the multiple hardware streams. In addition, each of the multiple hit/miss predictors is configured to forecast whether corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache. The system additionally includes a fetch algorithm that is coupled to the multiple hit/miss predictors. The fetch algorithm selects, *on a cycle-by-cycle basis,* the P of the multiple hardware streams from which to fetch the instructions.

In the rejection of claim 1, the Examiner notes that Yoaz has taught in a processor having multiple hardware streams supporting multiple date threads, and a date cache, a system for fetching individual ones of the multiple streams to a pipeline, comprising: a) multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said

corresponding one of the multiple hardware streams will hit or miss the data cache. The Examiner noted that col. 1, page 47, the paragraph beginning with "Another" clearly indicates that multiple threads exist, and a predictor is used for each thread. The Examiner noted that Yoaz teaches b) a fetch algorithm, configured to select, on a cycle-by-cycle basis, the selected one of the multiple hardware streams from which to fetch the instructions. Again referring Applicant to Yoaz page 47, the Examiner stated that the fetch algorithm includes the steps of fetching from a current thread as long as a miss is not predicted, and when a miss is predicted, switch to and fetching instructions from a second thread, noting that it should be realized that this happens on a cycle-by-cycle basis, as instructions are executed every cycle.

Applicant respectfully disagrees with the Examiner's basis for rejecting claim 1 for the following reasons. First, Yoaz teaches a hit/miss predictor in an *instruction scheduler* that predicts, *on a per-load basis*, whether a load instruction will his/miss a data cache and for *scheduling instructions for execution*. Applicant's invention as recited in claim 1 has nothing whatsoever to do with execution scheduling. Applicant's recited elements are related to instruction fetching. Clearly Yoaz does not apply, nor does Yoaz suggest anywhere in his article that such techniques could be applied in the fetch stage of a processor for determining from which stream to fetch instructions. In addition, Yoaz describes a hit/miss predictor in an instruction scheduler. Applicant, on the other hand, claims a fetch stage, for simultaneously fetching every cycle, instructions from up to the P hardware streams. Applicant further claims a fetch algorithm that is coupled to multiple hit/miss predictors. The fetch algorithm selects, *on a cycle-by-cycle basis,* the P hardware streams from which to fetch the instructions. Applicant has thoroughly searched the teachings of Yoaz to find any recitation, motivation, suggestion, or even a hint that the teachings therein could be applied to simultaneously fetching every cycle, instructions from up to P hardware streams. Yoaz utterly fails to describe any of the above noted elements or limitations.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection of claim 1.

With respect to claims 2-5 and 14, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by Yoaz, Parady, or Parady and Yoaz in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 2-5 and 14.

In a manner similar to claim 1, claim 6 recites, in combination with other elements, a fetch stage, for simultaneously fetching every cycle, instructions from up to P hardware streams; a plurality of hit/miss predictors, each configured to forecast whether corresponding instructions from a corresponding one of multiple hardware streams will hit or miss one or more of levels in a data cache; and a fetch algorithm, coupled to the multiple hit/miss predictors, configured to select, on a cycle-by-cycle basis, P hardware streams from which to fetch instructions. These elements and limitations are entirely absent from the instruction scheduler teachings of Yoaz. Neither does Parady suggest such elements or limitations. For reasons substantially noted above in arguments presented in traversal of the Examiner's rejection of claim 1, Applicant asserts with respect to the rejection of claim 6 that Yoaz teaches a technique to improve the scheduling of instructions for execution. Applicant's invention is directed towards improved instruction fetching in a multithreaded processor, that is, an entirely different application altogether.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection of claim 6.

With respect to claims 7-10, these claims depend from claim 6 and add further limitations that are neither anticipated nor made obvious by Yoaz, Parady, or Yoaz and Parady in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 7-10.

Like claims 1 and 6, claim 11 recites, a method for simultaneously fetching instructions every cycle from up to P hardware streams to a pipeline. The method includes, for each of the hardware streams, making a hit/miss prediction by a corresponding one of associated hit/miss predictors as to whether corresponding instructions for the each of the multiple hardware streams previously fetched will hit or miss the data cache; and

selecting, on a cycle-by-cycle basis, the P hardware streams from which to fetch the instructions. Neither Yoaz nor Parady suggests anything about these above noted elements. Consequently, for reasons substantially noted above in arguments presented in traversal of the Examiner's rejections of claim 1 and claim 6, Applicant asserts with respect to the rejection of claim 11 that Yoaz teaches how to more effectively schedule instructions *that have already been fetched into a pipeline,* for execution. Again, Applicant's invention is directed towards instruction simultaneous instruction fetching in a multithreaded processor.

Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 11.

With respect to claims 12-13 and 18-19, these claims depend from claim 11 and add further limitations that are neither anticipated nor made obvious by Yoaz, Parady, or Yoaz and Parady in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 12-13 and 18-19.

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claim 15 under 35 U.S.C. §103(a) as being unpatentable over Yoaz, as applied in the rejections under 35 U.S.C. §102 discussed above. Applicant respectfully traverses and notes that Yoaz does not teach the limitations and elements recited in claim 1. Nor does Yoaz in combination with Parady. Accordingly, since claim 15 adds further limitations over that recited in claim 1, it is respectfully requested that the rejection of claim 15 be withdrawn.

The Examiner rejected claims 16-17 and 20 under 35 U.S.C. §103(a) as being unpatentable over Yoaz, as applied in the rejections under 35 U.S.C. §102 discussed above, in view of Ryan, U.S. Patent No. 5,694,572. Applicant respectfully traverses and notes that Yoaz does not teach the limitations and elements recited in claims 1, 6, or 11, as noted in arguments provided above. Nor does Yoaz in combination with Parady. Accordingly, since claim 16 adds further limitations over that recited in claim 1, it is respectfully requested that the rejection of claim 16 be withdrawn. Likewise, since claim 17 adds further limitations over that recited in claim 6, it is respectfully requested that the

rejection of claim 17 be withdrawn. Furthermore, The Examiner rejected claim 15 under 35 U.S.C. §103(a) as being unpatentable over Yoaz, as applied in the rejections under 35 U.S.C. §102 discussed above. Applicant respectfully traverses and notes that Yoaz does not teach the limitations and elements recited in claim 1. Nor does Yoaz in combination with Parady. Accordingly, since claim 20 adds further limitations over that recited in claim 11, it is respectfully requested that the rejection of claim 20 be withdrawn.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-20 are in condition for allowance. Reconsideration of the rejections and consideration of the new claims are requested, and allowance of all claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone at the direct dial number provided if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: **EO 004 054 928 US**
DATE OF DEPOSIT: **2/22/05**
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop **PETITION**, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,
HUFFMAN PATENT GROUP, LLC

By _____
RICHARD K. HUFFMAN
Reg. No. 41,082
Tel.: (719) 575-9998

Date: 2/19/05

Attachment